



TITLE:

退化を導入したDifferential Evolutionにより最適化されたニューラルネットワークによる株価予測 (不確実性と意思決定の数理)

AUTHOR(S):

阪井, 節子; 高濱, 徹行

CITATION:

阪井, 節子 ...[et al]. 退化を導入したDifferential Evolutionにより最適化されたニューラルネットワークによる株価予測 (不確実性と意思決定の数理). 数理解析研究所講究録 2009, 1636: 79-87

ISSUE DATE:

2009-04

URL:

<http://hdl.handle.net/2433/140497>

RIGHT:

退化を導入した Differential Evolution により最適化された ニューラルネットワークによる株価予測

広島修道大学商学部

阪井 節子 (Setsuko Sakai)

Faculty of Commercial Sciences, Hiroshima Shudo University

広島市立大学大学院 情報科学研究科 高濱 徹行 (Tetsuyuki Takahama)

Department of Intelligent Systems, Hiroshima City University

1 はじめに

観測データなどの各種データに内在する変数間の関係、入出力関係を推定しようとするモデリングに関する研究が盛んに行われている。その中でも近年、ニューラルネットワークによるモデル推定の研究が多数行われ、さまざまな分野で応用されている。モデルは、モデルを記述するパラメータの数、パラメータの意味づけを規定するモデル構造、およびパラメータ値によって特徴付けられる。階層型ニューラルネットワークにおいては、階層数及び各階層のニューロン数がモデル構造に対応し、各ニューロンの結合荷重と閾値がパラメータ値に対応する。しかし、ニューラルネットワークを用いたモデル推定においては、次のような問題が指摘されている。(1) 適切なネットワーク構造を選択することが困難である。ネットワークが大きすぎると、モデルの汎化性能が低くなり、逆に、ネットワークが小さすぎると、学習が十分に行えない。しかし、多くの場合、適切なネットワーク構造に関する情報を利用することが出来ないため、試行錯誤的に最適なネットワーク構造を探索しなければならない。(2) 隠れ層の解釈が困難である。一般に、ニューラルネットワークでは、推定誤差を小さくするために、十分な数の隠れ層を用意する。学習された知識は、それら多数の隠れ層の各ユニットに分散して表現されたため、個々のユニットの意味づけが不明確となり、結果として学習された知識の解釈が困難となる。(3) 局所解に陥ることを回避できない。ニューラルネットワークを用いたモデル推定では、学習アルゴリズムとして最急降下法が用いられることが多く、ネットワークの大きさが大きくなるほど、この問題は深刻である。

これらの問題を解決するためには、結合荷重や閾値などのパラメータ値の最適化だけでなく、不必要な結合や閾などのパラメータ構造を最適化する必要がある。ニューラルネットワークの構造最適化手法は、選択法[1, 2]、削除法[3, 4, 5]、生成法[6, 7, 8]、縮退法[9]の4つの範疇に分類することが出来る。選択法は、幾つかのパラメータ構造において近似誤差が最小となるようにパラメータ値を最適化し、何らかの評価基準に基づいてその中から最適なものを選択する方法である。この方法では、AIC(Akaike Information Criterion)[10]、MDL(Minimum Description Length) Principle[11]、GPE(Generalized Prediction Error)[12]などが、評価基準として用いられる。削除法/生成法では、結合荷重のようなパラメータ値を学習する過程とユニットや結合のようなパラメータ構造を削除/追加する過程が、別々の過程として繰り返される。ネットワーク構造の変更が段階的に行われるため、ネットワーク構造の空間における山登り法による探索と考えられ、最適なネットワーク構造を探索することが困難である。さらに、通常、ネットワーク構造の変更前と変更後の乖離が大きく、結合荷重を再学習する必要がある、計算量が多くなるという問題点が指摘されている。縮退法は、パラメータ値を最適化する過程でパラメータを縮退させ、パラメータ数を削減する方法である。結合の削除過程が結合荷重の学習過程の中に含まれているため、結合の縮退が起こる直前には、縮退されるパラメータの値はほとんど0になっており、縮退の前後における構造の乖離が小さく、パラメータ値の再学習は不要である。

これに対して、我々は進化的アルゴリズム (Evolutionary Algorithm) に退化の概念を導入した次のような新たな構造最適化アルゴリズムを提案している。2値GAを用いたMGGA(Genetic Algorithm with Mutant Genes)[13]、実数値GAを用いたDGGA(Genetic Algorithm with Damaged Genes)[14, 15]、DGGAの拡張であるGA^d(Genetic Algorithm with Degeneration)[16, 17, 18, 19]、共進化を用いたCGA^d(Coevolutionary Genetic Algorithm with Degeneration)[20]、差分進化(DE)を用いたDE^d(Differential Evolution with Degeneration)[21]、等である。これらのアルゴリズムでは、遺伝子の損傷を用いて、個体の生存にあまり寄与していない遺伝子を削除したり不活性化することにより退化現象を実現し、モデル内の不要なパラメータを削除することにより、モデル構造の最適化を行う。このような構造最適化機能によって、前述の間

題 (1) および (2) は解決される。問題 (3) についても、EA は比較的局所解に陥りにくいため、解決が期待できる。これらのアルゴリズムは、多項式モデル、ニューラルネットワーク、RBF ファジィモデルなどの構造最適化に適用できることが示されている。

これらの退化を伴う EA では、各遺伝子を、遺伝子が全く損傷していない、すなわち完全に正常である、ときの状態を表す正常値と、その遺伝子がどの程度損傷しているかを示す損傷度の対で定義する。GA^dでは、遺伝的操作として単純一点交叉と Gauss 突然変異を採用している。一方、DE は、効果的かつ頑健なアルゴリズムと知られているが、遺伝的操作を単純な算術演算で実現するため、GA^dと同様な方法で退化現象を導入することが困難であった。そこで、正常値と損傷度を一体化する写像により表現型に変換し、遺伝的操作を行い、正常値と損傷度を分離する逆写像により遺伝子型に戻す仕組みを採用することによって退化現象を実現した、DE^dを提案している。

本研究では、DE^dを株価予測を行うニューラルネットワークの構造最適化に適用し、その有効性を検証する。さらに、DE と比較することによって、DE^dが、推定精度が高くかつ簡潔な構造を持つニューラルネットワークを得ることがきる構造最適化手法であることを示す。

以下、2. では退化を伴う遺伝的アルゴリズム GA^dを説明し、3. では、退化を伴う差分進化 DE^d、さらに、写像、遺伝的操作、逆写像について説明する。4. では、株価予測を行うニューラルネットワークの構造最適化に関する実験結果を示す。5. はまとめである。

2 退化を伴う遺伝的アルゴリズム (GA^d)

2.1 退化

自然界においては、進化の過程で不要な器官などを失う退化と呼ばれる現象が知られている。これは、その器官に関与する遺伝子が何らかの原因で損傷し、正常な遺伝子とは異なる損傷した遺伝子 (以下、損傷遺伝子と呼ぶ) となったものの、それが生存に不要な器官に対応する遺伝子であったために、個体が生き残り、損傷遺伝子が子孫に継承されたことによって起こった現象であると考えられる。モデルパラメータを器官と見なし、進化的アルゴリズム (EA) に対して退化の仕組みを導入することによって、多数のパラメータで記述されたシステムにおいて、不必要なパラメータを削除し、パラメータ数を最適化できることになる。

2.2 損傷遺伝子

損傷遺伝子は、突然変異により正常な遺伝子と異なる状態になった遺伝子である。突然変異には置換、挿入、欠損など様々な種類があるため、損傷遺伝子を取り得る全ての状態をあらかじめ想定することは困難である。そこで GA^dでは、遺伝子が正常であるときの状態を表す正常値 (normal value) とその遺伝子がどの程度損傷しているかを示す損傷度 (damaged rate) の対により、遺伝子表現する。損傷度は区間 $[0, 1]$ の値をとり、正常な遺伝子の損傷度は 0、形質を全く発現しなくなった遺伝子の損傷度は 1 とする。

GA では、各個体の持つ遺伝情報を伝える実体として染色体を仮定し、染色体を遺伝子の配列で表現する。GA における遺伝子配列を $G = g_1 g_2 \cdots g_L$ 、遺伝子型から表現型への写像を h 、適応度関数を f とする。ただし、 g_i は i 番目の遺伝子、 L は染色体の長さである。このとき、個体の適応度は $f(h(G))$ で与えられる。

GA^dでは、正常値の列が同じであっても損傷度によって各個体の表現型が変化する。GA^dにおける個体は以下の情報を持つ。

- 遺伝子の列 (G^d)

$G^d = (g_1, d_1)(g_2, d_2) \cdots (g_L, d_L)$ 。ただし、 g_i は第 i 番目の遺伝子の正常値、 d_i は損傷度を示している。GA^dでは、遺伝子は、ある確率で損傷を受け対応する損傷度が増加し、逆に、ある確率で修復され対応する損傷度が減少すると仮定する。

- 適合度

GA^dでは、遺伝子型から表現型への写像は、正常値 G だけでなく損傷度 D にも依存するため、個体の適応度は $f(h^d(G, D))$ で与えられる。通常、以下のような写像が用いられる。

$$h^d(G, D) = g_1 \cdot (1 - d_1) \cdots g_L(1 - d_L) \quad (1)$$

この写像では、ある遺伝子が完全に損傷している ($d_i = 1$) の場合、その遺伝子の表現型は 0 である。また、完全に正常である ($d_i = 0$) の場合、表現型は正常値 g_i となる。

図 1 に、 GA^d における個体の例を示す。

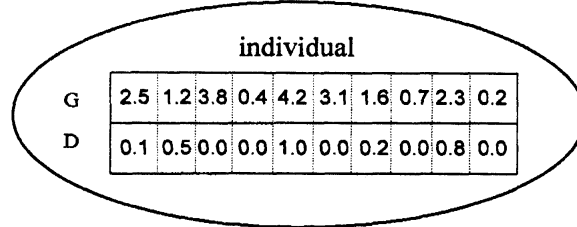


図 1: GA^d における個体の表現

2.3 GA^d のアルゴリズム

GA^d のアルゴリズムの概要を以下に示す。

- (0) 初期集団の生成：初期個体集団をランダムに生成する。通常、初期個体の損傷度は区間 $[0,1]$ でランダムに決めるが、すべて 0 あるいはすべて 1 とすることも可能である。
- (1) 選択：次世代の親となる個体を選択する。構造最適化では構造に対する評価関数を最小化することが多い。 GA^d では、頑健な方法として知られている線形ランキング選択 [22] を用いる。
- (2) 交叉：交叉確率 P_c で親の交叉を行い、子を生成する。正常値と損傷度の対が同時に子に継承される。なお、交叉しない場合には、親がそのまま子になったとみなす。
- (3) 可逆的突然変異：生成された子の正常値に対して、可逆的突然変異確率 P_{rm} で突然変異を起こす。正常値 g_i は次式にしたがって g_i^{new} に変化する。

$$g_i^{\text{new}} = g_i + \Delta g \quad (\text{ただし, } \Delta g \text{ は乱数}) \quad (2)$$

- (4) 不可逆的突然変異：生成された子の損傷度に対して、不可逆的突然変異確率 P_{im} で突然変異を起こす。損傷度が変化する際、損傷度が増加する確率は、現在の損傷度 $d \in [0,1]$ から確率 $p \in [0,1]$ への写像、損傷度増加確率関数 P_{dam}

$$P_{dam} : d \in [0,1] \rightarrow p \in [0,1] \quad (3)$$

によって $p = P_{dam}(d)$ で与えられる。不可逆的突然変異が起こったとき、損傷度は確率 p で増加し、確率 $1 - p$ で減少する。本研究では、損傷度の変化は次式で与えられる。

$$d_i^{\text{new}} = \begin{cases} d_i + \Delta d & \text{w.p. } P_{dam}(d_i) \\ d_i - \Delta d & \text{w.p. } 1 - P_{dam}(d_i) \end{cases} \quad (\text{ただし, } \Delta d \text{ は乱数}) \quad (4)$$

- (5) 世代交代：現在の集団を子に置き換え、(1)に戻る。

3 退化を伴う差分進化 (DE^d)

3.1 差分進化 (Differential Evolution: DE)

差分進化 (DE) は、Storn and Price[23, 24] によって提案された進化戦略 (evolution strategy) の一つであり、解集団による多点探索を行う確率的直接探索法である。非線形問題、微分不可能な問題、非凸問題、多峰性問題など、様々な最適化問題に適用され、高速で頑健なアルゴリズムであることが示されている。

DEには幾つかの形式が提案されており、DE/best/1/bin や DE/rand/1/exp などがよく知られている。これらは、DE/base/num/cross という記法で表現される。“base”は基本ベクトルとなる親の選択方法を指定する。例えば、DE/rand/num/cross は基本ベクトルのための親を集団からランダムに選択し、DE/best/num/cross は集団の最良個体を選択する。“num”は基本ベクトルを変異させるための差分ベクトルの個数を指定する。“cross”は子を生成するために使用する交叉方法を指定する。例えば、DE/base/num/bin は一定の確率で遺伝子を交換する交叉 (binomial crossover) を用い、DE/base/num/exp は、指数関数的に減少する確率で遺伝子を交換する交叉 (exponential crossover) を用いる。

DEでは、探索空間中にランダムに初期個体を生成し、初期集団を構成する。各個体は決定ベクトルに対応し、 n 個の決定変数を遺伝子として持つ。各世代において、全ての個体を親として選択し、各親に対して、次のような処理が行われる。現在、親として選択された個体を除く個体群から互いに異なる $1+2\text{ num}$ 個の個体を選択する。最初の個体が基本ベクトルとなり、残りの個体対が差分ベクトルとなる。差分ベクトルはスケールリングファクタ F (scaling factor) が乗算され基本ベクトルに加算される。得られたベクトルと親を、交叉率 CR (crossover rate) に基づいて交叉し、子ベクトル (trial vector) を生成する。最後に、生存者選択として、子が親よりも良ければ、親を子で置換する。

本研究では、差分ベクトル数を 1 ($\text{num} = 1$) とした DE/rand/1/exp を用いる。

3.2 表現型への写像と逆写像、遺伝的操作

GA^d では、正常値と損傷度を対として扱ったが、DEは遺伝的操作を単純な算術演算で実現するため、 GA^d と同様な方法で退化現象を導入することは困難である。そこで、正常値と損傷度を一体化する写像により表現型に変換し、遺伝的操作を行い、正常値と損傷度を分離する逆写像により遺伝子型に戻す、という簡単な方法を提案した。

2つの親個体を $G_i^d = (g_{i1}, d_{i1}) \cdots (g_{iL}, d_{iL})$, $i = 1, 2$ と仮定する。

1. 写像：遺伝的操作をする前に、親の正常値と損傷度を一体化し、表現型に写像する。遺伝子毎の写像関数は、 $h^d(g_j, d_j) = g_j(1 - d_j)$ であり、表現型は次式で与えられる。

$$H_i^d = g_{i1}(1 - d_{i1}) \cdots g_{iL}(1 - d_{iL}), \quad i = 1, 2. \quad (5)$$

2. 遺伝的操作： H_1^d と H_2^d に対して算術交叉のような遺伝的操作を適用し、 $H_i^{d'}$ ($i = 1, 2$) を生成する。

$$H_i^{d'} = h'_{i1} \cdots h'_{iL}, \quad i = 1, 2. \quad (6)$$

さらに、 D_1 と D_2 に対して 1 点交叉または一様交叉のような交叉を適用し、 $D_i' (= 1, 2)$ を生成する。

$$D_i' = d'_{i1} \cdots d'_{iL}, \quad i = 1, 2. \quad (7)$$

3. 逆写像： D_i' を用いる逆写像を用いて $H_i^{d'}$ から正常値を求め、新しい個体 $G_i^{d'}$ を生成する。遺伝子毎の逆写像関数は、 $h^{d^{-1}}(h_j, d_j) = h_j / (1 - d_j)$ である。

$$G_i^{d'} = (g'_{i1}, d'_{i1}) \cdots (g'_{iL}, d'_{iL}) \quad (8)$$

$$g_{ij}' = \begin{cases} h_{ij}' / (1 - d_{ij}'), & d_{ij}' \neq 1 \\ 0, & d_{ij}' = 1 \end{cases} \quad (9)$$

遺伝子が完全に損傷して損傷度が 1 であるときは、0 による除算を避けるために正常値を 0 とする。

これらの写像、遺伝的操作、逆写像の連続的な操作が、GAにおける交叉に対応する。これらの操作を行った後、必要ならば正常値に対して可逆的突然変異を適用し、損傷度に対して不可逆的突然変異を適用する。

3.3 DE^dのアルゴリズム

DE^dのアルゴリズムの概要を以下に示す.

- (0) 初期集団の生成：初期個体集団 $\{G_i^d = (G_i, D_i)\}$ をランダムに生成する. 初期個体の損傷度については, 通常区間 $[0,1]$ でランダムに決める.
- (1) 終了判定：世代数が最大世代数 T_{\max} に達したとき, 実行を終了する.
- (2) 写像：各個体 G_i^d に対して, 基本ベクトル G_{p1}^d , 差分ベクトルのための G_{p2}^d, G_{p3}^d を G_i^d および互いに重複しないようにランダムに選択する. これら 4 個すべての個体を, 各遺伝子に対する写像 $h^d(g_j, d_j) = g_j(1 - d_j)$ によって表現型に変換し, $H_i, H_{pk}, k = 1, 2, 3$ を得る.
- (3) 遺伝的操作：新しいベクトル H' を基本ベクトル H_{p1} および差分ベクトル $H_{p2} - H_{p3}$ から以下のよう生成する.

$$H' = H_{p1} + F(H_{p2} - H_{p3}) \quad (10)$$
 H' を親 H_i と交叉し, ベクトル H'' を生成する. さらに, 親の損傷度 D_i と基本ベクトルの損傷度 D_{p1} を交叉し, D'_i を得る.
- (4) 逆写像： D'_i を用いた逆写像 $h^{d^{-1}}(g_j, d_j) = h_j/(1 - d_j)$ によって, H'' から正常値 G'_i を求め, 新しい個体 $G_i^{d'} = (G'_i, D'_i)$ を生成する.
- (5) 不可逆的突然変異：損傷度に対して, 式 (3) による突然変異を起こす.
- (6) 生存選択：子ベクトルが親ベクトルより良ければ, 親を子に置き換える
- (7) (1)に戻る.

4 ニューラルネットワークの構造最適化

本節では, ニューラルネットワークの表現とその構造最適化について説明する.

4.1 コーディング

染色体 G^d を次のような階層型ニューラルネットワークにより記述する.

$$G^d = L^2 L^3 \dots L^m \quad (11)$$

$$L^k = N_1^k N_2^k \dots N_{n^k}^k \quad (12)$$

$$N_i^k = (w_{i1}^{k1} d_{i1}^{k1}) \dots (w_{in^{k-1}}^{k, n^{k-1}} d_{in^{k-1}}^{k, n^{k-1}}) (\theta_i^k d_{i\theta}^k) \quad (13)$$

ここで, L^k は第 k 層, m はネットワークの階層数, N_i^k は層 L^k の第 i ニューロン, n^k は層 L^k に含まれるニューロン数, $(w_{ij}^{k1} d_{ij}^{k1})$ はニューロン N_i^k と N_j^{k-1} の結合加重, $(\theta_i^k d_{i\theta}^k)$ はニューロン N_i^k の閾値である. 染色体 G^d から, 次の結合荷重と閾値が得られる.

$$w_{ij}^k = w_{ij}^{k1}(1 - d_{ij}^{k1}), \quad \theta_i^k = \theta_i^{k1}(1 - d_{i\theta}^{k1}) \quad (14)$$

4.2 適合度

第 p 入力パターン I^p ($p = 1, 2, \dots, P$) に対する, ニューロン N_i^k の出力 O_i^k は次式で与えられる.

$$O_i^k(I^p) = f\left(\sum_j w_{ij}^k O_j^{k-1}(I^p) - \theta_i^k\right) \quad (15)$$

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (16)$$

ここで、 $f(\cdot)$ は出力関数である。

適合度は、式 (17) にあるように訓練データとの平均平方誤差 (MSE) で定義し、適合度を最小化した。

$$MSE = \frac{1}{P} \sum_p \sum_i (\hat{O}_i^p - O_i^m(I^p))^2 \quad (17)$$

ここで、 P は入力パターン数、 \hat{O}_i^p は第 p 入力パターン I^p に対する教師信号の第 i 成分である。

4.3 株価予測

本研究では、株価指数日経 225 を予測する。日経 225 の時系列データ X を次式で表す。

$$X = \{x(1), x(2), \dots, x(t), \dots\} \quad (18)$$

テクニカル分析では、短期の変動を平滑化し長期傾向あるいは循環変動を明確にするために、時系列データの移動平均値がしばしば用いられる。 m 日間の移動平均値 $MA(t, m)$ は、 m 日間の株価の算術平均によって与えられる。

$$MA(t, m) = \frac{x(t) + x(t-1) + \dots + x(t-m+1)}{m} \quad (19)$$

本研究では、短期傾向を知るために 5 日間の移動平均 (5DMA)、中期傾向を知るために 25DMA と 75DMA、長期傾向を知るために 100DMA をそれぞれ用いる。株価 $x(t)$ の予測には、次の 8 項目を使用する。

$$\begin{aligned} &x(t-1), x(t-2), x(t-3), x(t-4), \\ &MA(t-1, 5), MA(t-1, 25), MA(t-1, 75), MA(t-1, 100) \end{aligned} \quad (20)$$

実験には、4 層の階層型ニューラルネットワークを用い、入力層、2 層、3 層のニューロン数はそれぞれ 8 個、出力層のニューロン数は 1 個とした。したがって、学習パラメータ (結合加重と閾値) の総数は 153 である。

訓練データは 1,229 件 (2003 年 1 月 6 日から 2007 年 12 月 28 日までの日経 225 の終値) である。図 2 に、この期間における訓練データ及び式 (20) であげられた 8 項目の推移を示す。なお、これらの値は、次式によって区間 $[0, 1]$ の範囲の値に正規化した。

$$x(t)' = \frac{x(t) - 6000}{14000} \quad (21)$$

一方、検査データは 147 件 (2008 年 1 月 4 日から 2008 年 8 月 6 日の間の日経 225 終値) である。

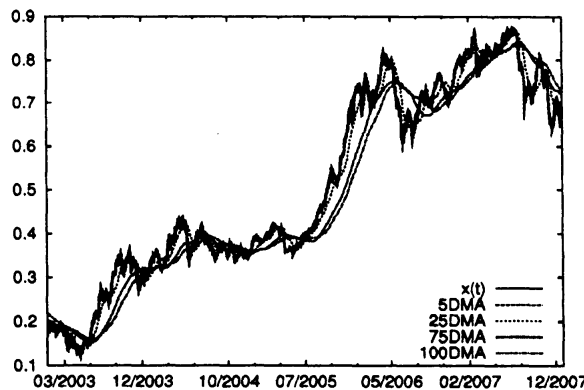


図 2: 正規化された訓練データ (2003 年 1 月 6 日から 2007 年 12 月 28 日)

4.4 実験結果

実験条件は、次のとおりである。

- 共通条件：個体数 $N = 100$, DE/rand/1/exp, スケーリングファクタ $F = 0.8$, 交叉率 $CR = 0.95$ とする。初期集団の正常値は区間 $[-5, 5]$ の一様乱数とし、可逆的突然変異は用いない。
- DE^d に関する条件：初期個体の損傷度は区間 $[0, 1]$ の乱数とした。損傷度は、式 (3), (4) による不可逆的突然変異によって変化する。 Δd は、正規分布 $N(0, 0.2^2)$ による正規乱数とする。表現型への写像, 正常値への逆写像を用い、損傷度については一様交叉を用いる。不可逆的突然変異確率 $P_{im} = 1/L$, 損傷度増加関数は定数関数 $P_{dam}(d) = 0.9$ とする。

最大世代数 $T_{max} = 500$ とし、評価関数値としては 10 回の試行の平均値を用いた。

DE^d と DE の実験結果を表 1 に示す。Min, Mean, Max, Std. はそれぞれ、各試行における最良個体による訓練データと検査データそれぞれに対する平均平方誤差 (MSE) の最小値, 平均値, 最大値, 標準偏差である。最良個体のパラメータ数は、 DE^d で 145.2, DE で 153 であり、 DE^d は DE より精度が高く、簡潔な構造のネットワークを得ている。

表 1: DE^d と DE の比較

Data	Item	DE^d	DE
Training Data	Min.	3.365e-04	3.769e-04
	Mean	4.207e-04	6.551e-04
	Max.	5.822e-04	9.553e-04
	Std.	7.596e-05	1.782e-04
Test Data	Min.	5.965e-04	5.109e-04
	Mean	8.106e-04	1.658e-03
	Max.	1.229e-03	4.808e-03
	Std.	2.190e-04	1.158e-03

図 3 に、正規化された訓練データ, DE^d による 10 回の試行で得られた最良個体のメジアンを用いて得られた訓練データの予測値, DE による予測値を示す。この結果から、 DE^d は DE より訓練データに近い値を予測できていることが分かる。

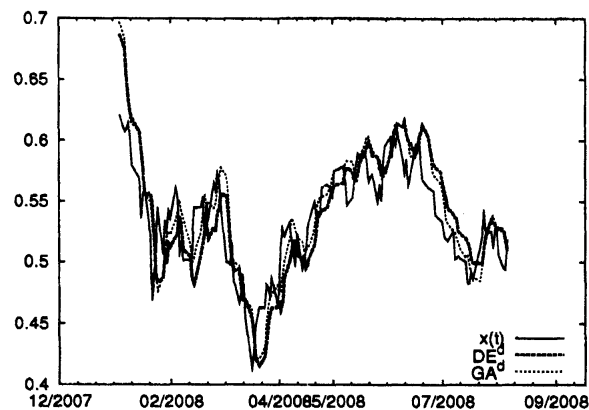


図 3: DE^d と DE による訓練データの予測値 (2008 年 1 月 4 日から 8 月 6 日)

5 終わりに

本研究では、進化的アルゴリズムに対して表現型への写像と正常値への逆写像を用いることによって退化を実現する方法を提案し、効果的かつ頑健なアルゴリズムとして知られる差分進化に対して適用し、退化を伴う差分進化 DE^d を提案した。 DE^d を、株価指数日経 225 の株価予測を行うニューラルネットワークの構造最適化に用い、 DE に比べて精度が高く簡潔な構造のニューラルネットワークが得られることを示した。この結果より、 DE^d は DE に比べて、高い精度と簡潔な構造を両立する最適化ができる良好な構造最適化アルゴリズムであると考えられる。

今後は、 DE^d をファジィモデルなどの他のモデルへの適用を進めるとともに、その他の様々な分野への応用を試みる。さらに、より安定的に優れた構造を発見するために、退化圧力の動的制御を導入することについても検討する。

謝辞

本研究の一部は、日本学術振興会科学研究費補助金基盤研究 (C) (2) (課題番号 17510139, 20500138) 及び広島市立大学特定研究費 (一般研究)7111 による補助を頂いた。

参考文献

- [1] T. Kurita: "A method to determine the number of hidden units of three layered neural networks by information criteria", Trans. of the Institute of Electronics, Information and Communication Engineers, J73-D-II, 11, pp. 1872-1878 (1990). In Japanese.
- [2] N. Murata, S. Yoshizawa and S. Amari: "Network information criterion-determining the number of hiddenunits for an artificial neural network model", IEEE Trans. on Neural Networks, 5, 6, pp. 865-872 (1994).
- [3] Y. L. Cun, J. S. Denker and S. A. Solla: "Optimal brain damage", Advances in Neural Information Processing Systems 4 (Ed. by D. S. Touretzky), Morgan Kaufmann, San Mateo, CA, pp. 598-605 (1992).
- [4] B. Hassibi and D. Stork: "Second Order Derivatives for Network Pruning: Optimal Brain Surgeon", Advances in Neural Information Processing Systems (Eds. by J. D. C. S. J. Hanson and C. L. Giles), Vol. 5, Morgan Kaufmann, San Mateo, pp. 164-171 (1993).
- [5] M. C. Mozer and O. Smolensky: "Using relevance to reduce network size automatically", Connection Science, 1, 1, pp. 3-16 (1989).
- [6] C. Campbell: "Constructive learning techniques for designing neural network systems", Neural Network Systems Technologies and Applications (Ed. by C. T. Leondes), Academic Press, San Diego (1997).
- [7] S. E. Fahlman and C. Lebiere: "The cascade-correlation learning architecture", Advances in Neural Information Processing Systems 2 (Ed. by D. S. Touretzky), Morgan Kaufmann, San Mateo, CA, pp. 642-649 (1992).
- [8] R. Parekh, J. Yang and V. Honavar: "Constructive neural-network learning algorithms for pattern classification", IEEE Trans. on Neural Networks, 11, 2, pp. 436-451 (2000).
- [9] M. Ishikawa: "Structural learning with forgetting", Neural Networks, 9, 3, pp. 509-521 (1996).

- [10] H. Akaike: "A new look at the statistical model identification", *IEEE Trans. Automatic Control*, **AC-19**, 6, pp. 716–723 (1974).
- [11] J. Rissanen: "Stochastic Complexity and Modeling", *The Annals of Statistics*, **14**, 3, pp. 1080–1100 (1986).
- [12] J. Moody: "The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems", *Advances in Neural Information Processing Systems* (Eds. by J. E. Moody, S. J. Hanson and R. P. Lippmann), Vol. 4, Morgan Kaufmann, San Mateo, pp. 847–854 (1992).
- [13] 高濱, 阪井, 磯道: "変異遺伝子を導入した遺伝的アルゴリズム (MGGA) の提案", *電子情報通信学会論文誌*, **J84-D-I**, 9, pp. 1297–1306 (2001).
- [14] T. Takahama and S. Sakai: "Structural optimization of neural network by genetic algorithm with damaged genes", *Proc. of the 9th International Conference of Neural Information Processing (ICONIP'02)*, Vol. 3, pp. 1211–1215 (2002).
- [15] T. Takahama and S. Sakai: "Structural learning by genetic algorithm with damaged genes", *Proc. of the IASTED International Conference on Artificial and Computational Intelligence (ACI2002)*, ACTA Press, Anaheim, USA, pp. 161–166 (2002).
- [16] 高濱, 阪井, 市村, 磯道: "退化現象を導入した遺伝的アルゴリズム GA^d による構造最適化", *電子情報通信学会論文誌*, **J86-D-I**, 3, pp. 140–149 (2003).
- [17] T. Takahama, S. Sakai and Y. Isomichi: "Structural optimization of neural networks by genetic algorithm with degeneration (GA^d)", *Neural Information Processing: Research and Development* (Eds. by J. C. Rajapakse and L. Wang), Springer, pp. 256–277 (2004).
- [18] T. Takahama, S. Sakai and N. Iwane: "Learning structure of rbf-fuzzy rule bases by degeneration", *Proceedings of 2003 International Conference on Fuzzy Information Processing (FIP2003)*, pp. 611–616 (2003).
- [19] T. Takahama, S. Sakai and N. Iwane: "Structural learning of rbf-fuzzy rule bases on information criteria and degeneration", *Proc. of 2003 IEEE International Conference on Systems, Man, and Cybernetics (SMC2003)*, pp. 2581–2586 (2003).
- [20] T. Takahama and S. Sakai: "Structural learning of neural networks by coevolutionary genetic algorithm with degeneration", *Proc. of 2004 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3507–3512 (2004).
- [21] T. Takahama, S. Sakai, A. Hara and N. Iwane: "Structural learning of neural networks by differential evolution with degeneration using mappings", *Proc. of the 2007 IEEE Congress on Evolutionary Computation*, pp. 3434–3441 (2007).
- [22] J. E. Baker: "Adaptive selection methods for genetic algorithms", *Proc. of the First International Conference on Genetic Algorithms and Their Applications*, Lawrence Erlbaum Associates, pp. 101–111 (1985).
- [23] R. Storn and K. Price: "Minimizing the real functions of the ICEC'96 contest by differential evolution", *Proc. of the International Conference on Evolutionary Computation*, pp. 842–844 (1996).
- [24] R. Storn and K. Price: "Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces", *Journal of Global Optimization*, **11**, pp. 341–359 (1997).